Traffic Modeling and Classification Using Packet Train Length and Packet Train Size

Dinil Mon Divakaran[1], Hema A. Murthy[1], and Timothy A. Gonsalves[1]

Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Chennai - 600036
{dinil,hema,tag}@tenet.res.in

Abstract. Traffic modeling and classification finds importance in many areas such as bandwidth management, traffic analysis, traffic prediction, network planning, Quality of Service provisioning and anomalous traffic detection. Network traffic exhibits some statistically invariant properties. Earlier works show that it is possible to identify traffic based on its statistical characteristics. In this paper, an attempt is made to identify the statistically invariant properties of different traffic classes using multiple parameters, namely packet train length and packet train size. Models generated using these parameters are found to be highly accurate in classifying different traffic classes. The parameters are also useful in revealing different classes of services within different traffic classes.

1 Introduction

The phenomenal expansion of Internet has seen a rapid growth in the number and variety of applications. Many of such applications turn out to be bandwidth-hungry or delay-sensitive; and require, or at least benefit from specific service classes that prioritize packets in the Internet. Internet Service Providers and network operators need to classify traffic data within their network and evaluate their absolute and relative importance and subsequently create traffic policies to be enforced in the Internet routers [1]. Network administrators need to know the different classes or types of traffic that flow through their network so as to manage the network efficiently. Accurate traffic classification is essential for provisioning and bandwidth management. Floyd and Paxson [2] pointed out that it is important to capture the invariants of traffic to cope with the constantly changing nature of Internet traffic.

Conventional methods for traffic classification use the packet header information to find out the ports used for communication. Well known ports are supposed to be used by specific application protocols (e.g. port 80 is usually used by HTTP). But this method has become less and less accurate as more and more emerging applications use well know ports for relaying traffic, e.g. tunneling over HTTP port is very common [3, 4]. Owing to the increasing traffic and application protocols, techniques based on statistical modeling are gaining

importance. Recent works have made efforts to increase the accuracy in traffic classification [5–8].

In traffic modeling, parameters are extracted from packet headers. There are different parameters such as packet length, packet inter-arrival time, flow duration, packet train inter-arrival time, packet train length, packet train size etc that can be considered for modeling of traffic. While parameters can be modeled separately [9], we focus on modeling the traffic classes using multiple parameters.

In [10], it was shown that traffic characteristics are generally multimodal in nature. For example, the total number of packets transferred during mail transfer vary for small text messages to large picture attachments. To capture the multimodal characteristic of traffic, we employ clustering techniques based on Vector Quantization (VQ) [11] and Gaussian Mixture Models (GMM) [12]. The models obtained using these techniques are later used for classification of a given data set (not used during training) into one of the different traffic types.

The rest of the paper is organized as follows. In Sect.3 we explain the experimental setup. Modeling and classification using VQ is explained in Sect.4. Section 5 details modeling using GMM and Bayesian classification using the Gaussian mixtures obtained. Evaluation and verification of models follow in Sect.6. We conclude in Sect.7.

2 Related Work

System administrators have long being using port based classification method to identify different traffic classes flowing in the network. Tools such as tcpdump [13] read the header information to find the source and destination ports. Each server port associates itself with an application as per the IANA (Internet Assigned Numbers Authority) [14], which maintains a mapping of the server ports to application types. Such a method does not provide good accuracy for a number of reasons. For instance, with the proliferation of applications, not every application is registered with IANA. Users behind a firewall that permits packets to only a few ports, usually relay traffic through well known ports (eg. SSH over HTTP). Similarly, non-privileged users run HTTP servers on ports other than 80. Even for applications defined with IANA, some ports are used by different applications with entirely different QoS requirements (for example, SSH and SCP use same port 22).

Statistical traffic classification is an alternative to the less accurate port based classification. Past research works have focused on characterizing particular traffic classes. In [15], joint distribution of flow duration and number of bytes were used to identify DNS traffic. Paxson [16] examined the distribution of flow bytes and packets for a number of different applications. Roughan et al. used LDA (Linear Discriminant Analysis) and QDA (Qualitative Discriminant Analysis) to classify traffic into different classes of services for QoS implementation [5]. In [6], authors describe a method for visualisation of the attribute statistics that aids in recognizing cluster types. The method uses EM (Expectation-Maximistaion)

for probabilistic clustering with parameters such as packet size and packet inter-arrival time. In [8], the authors explore Bayesian classifier using a number of per flow discriminators to classify network traffic.

The multimodal nature of traffic was highlighted in [9] using packet size as a promising parameter for traffic characterisation. We extend this work and use flow related information for modeling traffic.

3 Traffic modeling

In this section, we detail the setup for experimentation in terms of the parameters used for identifying the various traffic classes considered for this work.

3.1 Parameters

The goodness of these models largely depends on the parameters that are used for modeling. The parameters used here for modeling and classification of traffic classes are *packet train length* and *packet train size*. The concept of packet train was introduced by Raj Jain and Shawn A. Routhier [17]. The two ends of a packet train are identified as two nodes in a network. As defined in [17], a packet train is essentially the flow of packets between two nodes in a network, where each packet forms the car of the train. Here, we modify the definition of a packet train to be the flow between two sockets, which is appropriately identified by the quadruple *(source host, source port, destination host, destination port)*. Each end of such a packet train is identified by the node name and port number. Packet train length is then defined as the number of packets within a train; and packet train size is the sum of the sizes of all the packets that form a packet train. The advantage of using these parameters (as will be discussed later), is that the models generated using these parameters give information regarding the class of service within an application type (say HTTP), apart from classifying traffic.

Since we consider two parameters for modeling and testing, it is important to ensure that one parameter doesn't overshadow the other parameter. Appropriate normalization of parameters is therefore performed.

3.2 Traffic classes

Five commonly used application protocols are selected for modeling and testing. These are HTTP, SMTP, DNS, SSH and POP3 which use ports 80, 25, 53, 22 and 110 respectively. DNS uses UDP, whereas HTTP, SMTP, SSH and POP3 use TCP as the transport layer protocol. The terms *traffic types* and *traffic classes* refer to these application protocols in general.

This work looks only at one UDP based traffic class, namely DNS traffic. Since all other traffic classes considered are TCP based, and hence connection oriented, they inherently have the packet train property. But, it should be noted that almost all UDP based applications can be viewed as a connection oriented traffic and therefore can be represented using packet trains. For example, a video

conferencing tool running on top of UDP can be identified uniquely by *src host*, *src port*, *dst host* and *dst port*, and all such packets can be considered as part of a connection. To distinguish different connections of the same applications, the time between packets can be used; as the time between consecutive packets of one run of the application will be much less as compared to time between consecutive packets of different runs of the application. Hence, using all these information, packet train parameters can be extracted for the traffic generated by a video conferencing application, or in general, for traffic generated by almost any UDP based application.

3.3 Data Representation

Throughout the paper, data or packet trains are represented as a set of N vectors, $\boldsymbol{X} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\}$ [1]. Each input vector has a dimension of two corresponding to the two parameters used in modeling.

4 Classification using Vector Quantization

VQ is a very popular approximate method in the class of clustering algorithms that simplifies computations and accelerates convergence. VQ partitions d-dimensional vectors in the vector space, $\mathbf{R}^d$, into finite sets of vectors based on the *nearest-neighbour* criterion [18]. Such sets, called clusters, represent separate regions in the vector space. A vector, $\mathbf{x} \in \mathbf{R}^d$, belongs to cluster C_i, if

$$\|\mathbf{x} - \boldsymbol{\mu_i}\| < \|\mathbf{x} - \boldsymbol{\mu_j}\| \quad \text{for all } j \neq i \ . \tag{1}$$

where $\boldsymbol{\mu_i}$ is the mean vector of the cluster, C_i. This equation states that a vector belongs to the nearest cluster. If there are two or more clusters to which the distance from the vector is minimum, one among them is chosen randomly. The clusters partition the vector space such that

$$\bigcup_{i=1}^{k} C_i = \mathbf{R}^d \qquad and \qquad \bigcap_{i=1}^{k} C_i = \phi \ . \tag{2}$$

where k is the number of clusters.

4.1 Training

During training, we use VQ to partition the vector space, where the first dimension of each vector is *packet train length* and second dimension is *packet train size*. The algorithm used is as follows

1. Initialize the mean (vector) of each cluster by randomly selecting a vector from the given set of vectors, $\mathbf{X}$, such that no two clusters have the same mean.

[1] Boldface is used to denote vectors and matrices.

2. Until the mean of each cluster converges
   - Classify each vector into one of the clusters using (1).
   - Recompute the mean of each cluster.

Using the above algorithm, the vectors in the given data set, $\mathbf{X}$, are classified into clusters. The models thus generated for each traffic type is used for testing.

4.2 Testing

In the testing phase, we have to find the traffic type that is *nearest* to the given data set, $\mathbf{X}$. For this, we calculate the distance between the i^{th} vector, $\mathbf{x_i}$, and the nearest cluster (best cluster) of a particular traffic type, s, using the Euclidean distance function

$$dist(\mathbf{x}_i, s) = \min_j \sqrt{\sum_{l=1}^{d} \left[\mathbf{x_{i}}_l - \boldsymbol{\mu}_{\mathbf{j}l}\right]^2} \qquad 1 \leq j \leq n(s) \ . \tag{3}$$

where $n(s)$ is the number of clusters in the traffic type s, and d is the dimension of the vectors. It should be noted that, since the clusters depend on the traffic type, the mean vector, $\boldsymbol{\mu}_\mathbf{j}$, for each cluster is also dependent on the traffic type. Now, the distortion of the data set, $\mathbf{X}$, from each traffic type is given by

$$D(s) = \sum_{i=1}^{N} dist(\mathbf{x_i}, s) \qquad 1 \leq s \leq S \ . \tag{4}$$

where S is the number of traffic types. The traffic type, T, of the given data set is identified as the one to which the distortion is minimum,

$$T = \arg\min_s D(s) \ . \tag{5}$$

5 Bayesian Classification using Gaussian Mixtures

Traffic classes can also be modeled using GMM. If each dimension of a d-dimensional vector $\mathbf{x}$ is normally distributed random variable with its own mean and variance, and are independent, their joint density has the form

$$p(\mathbf{x}) = \prod_{i=1}^{d} p(x_i) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}((x_i-\mu_i)/\sigma_i)^2} \ . \tag{6}$$

The multivariate Gaussian distribution function [19] for a d-dimensional vector $\mathbf{x}$ is written as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \ . \tag{7}$$

where $\boldsymbol{\mu}$ is the d-component *mean vector* and $\boldsymbol{\Sigma}$ is the d-by-d *covariance* matrix [2]. As the d components of the vector are independent, the covariance matrix reduces to a diagonal matrix,

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 0 & 0 & \dots & \sigma_d^2 \end{bmatrix} \quad . \tag{8}$$

$\sigma_1, \sigma_2, .., \sigma_d$ being the standard deviation along each of the vector component. In our work, the components of the vectors are nothing but the packet train parameters which are statistically independent.

The number of mixtures in a traffic type is a function of the traffic class s, denoted as $n(s)$. Let $\theta_{\mathbf{1}}, \theta_{\mathbf{2}}, ..., \theta_{\mathbf{n(s)}}$ be the vectors corresponding to the mixtures $m_1, m_2, ..., m_{n(s)}$, where $\theta_{\mathbf{i}}$ is the vector with components $\boldsymbol{\mu}_{\mathbf{i}}$ and $\boldsymbol{\Sigma}_{\mathbf{i}}$ of the mixture m_i. Given the feature vector, $\mathbf{x}$, the Bayes formula [19] to determine the (posteriori) probability of $\mathbf{x}$ being in the i^{th} mixture, m_i, is given as

$$P(\theta_{\mathbf{i}}|\mathbf{x}) \approx P(m_i) p(\mathbf{x}|\theta_{\mathbf{i}}) \quad . \tag{9}$$

An estimate of the probability of a mixture (prior) is calculated as

$$P(m_i) = \frac{n_{m_i}}{N} \quad . \tag{10}$$

where n_{m_i} is the number of vectors in m_i, and N is the total number of vectors.

The Bayes formula can be re-written by substituting the Gaussian function for the *likelihood* of m_i with respect to $\mathbf{x}$

$$P(\theta_{\mathbf{i}}|\mathbf{x}) \approx P(m_i) \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_{\mathbf{i}}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{i}})^t \boldsymbol{\Sigma}_{\mathbf{i}}{}^{-1} (\mathbf{x}-\boldsymbol{\mu}_{\mathbf{i}})} \quad . \tag{11}$$

where $\boldsymbol{\mu}_{\mathbf{i}}$ is the mean vector and $\boldsymbol{\Sigma}_{\mathbf{i}}$ is the covariance matrix of mixture m_i.

5.1 Training

We use (9) to determine the probability for each vector in the data set. The parameters (dimensions of the vector) are *packet train length* and *packet train size*. The following algorithm is used in the training phase.

1. Initialize the prior probability of each mixture with equal value such that their sum is 1.
2. Initialize the mean (vector) of each mixture by selecting a vector randomly, such that no two mixtures have the same mean.

[2] superscript t denotes the transpose, and $|\boldsymbol{\Sigma}|$ and $\boldsymbol{\Sigma}^{-1}$ are the determinant and the inverse of the covariance matrix respectively

3. Initialize the covariance matrix of each mixture to the d-by-d identity matrix.
4. Until the mean and variance of mixtures converge
   - For each vector in the given data set, classify it into mixture m_i if

$$P(\theta_{\mathbf{i}}|\mathbf{x}) > P(\theta_{\mathbf{j}}|\mathbf{x}) \quad \text{for all } j \neq i \ . \tag{12}$$

     where $\theta_{\mathbf{i}}$ corresponds to mixture m_i. If there are two or more mixtures with maximum probabilities, one among them is chosen arbitrarily.
   - Recompute the probability of each mixture.
   - Recompute the mean vector of each mixture.
   - Recompute the covariance matrix of each mixture.

The above algorithm basically computes the vector $\theta_{\mathbf{i}}$ and the probability corresponding to every mixture of a traffic class.

5.2 Testing

The testing phase uses the probabilities and the means and variances of mixtures of different traffic classes obtained from the training phase. In Bayesian clustering, the probability that the i^{th} vector, $\mathbf{x_i}$ belongs to a traffic class s is found using

$$P(s|\mathbf{x_i}) = \sum_j P(\theta_{\mathbf{j}}|\mathbf{x_i}) \qquad 1 \leq j \leq n(s) \ . \tag{13}$$

where $n(s)$ is the number of mixtures in the traffic class denoted by s.

We have observed that the rule which includes only the maximum probability corresponding to the best mixture, yields better results as compared to the above rule. We have therefore chosen the probability of occurrence of a vector in a traffic class to be the maximum of the probabilities of occurrence of the vector in each of the mixtures in the particular traffic class. That is

$$P(s|\mathbf{x_i}) = \max_j P(\theta_{\mathbf{j}}|\mathbf{x_i}) \qquad 1 \leq j \leq n(s) \ . \tag{14}$$

The probability that the given set of vectors belong to a particular traffic class is determined by the joint probability of all the vectors

$$P(s) = \prod_{i=1}^{N} P(s|\mathbf{x_i}) \ . \tag{15}$$

where N is the total number of vectors

Traffic classes overlap and the posterior probabilities obtained for mixtures across different traffic classes can be similar. We therefore apply a threshold on the probabilities (obtained by experimentation) before assigning a particular class. The thresholds are different for each of the traffic classes. For example, POP3 and SMTP have similar characteristics. A threshold based classification reduces the misclassification of POP3 as SMTP. The traffic class of the given data set is based on the following decision:

$$T = \arg\max_s \left[P(s) : P(s) > \alpha(s) \right] \ . \tag{16}$$

where $\alpha(s)$ is the threshold for traffic class, s.

6 Evaluation

The packet traces were collected from a gateway connecting the TeNeT [20] private LAN and Internet Service Provider's 1 Mbps (megabits per second) link, for a period of 90 days using tcpdump [13]. The packet header information from the tcpdump output is fed to a parser program. The parser generates data sets by computing the packet train parameters of each traffic type based on packet headers information (such as source IP address, destination IP address, source port, destination port, packet length and TCP flags). A data set used during training consists of packet train parameters of a single traffic type. The training was performed on data taken from 60 days of real Internet traffic. Data sets for varying time periods of 60 minutes, 30 minutes and 15 minutes were used for training. This essentially means that one-hour model for the i^{th} hour has packet train information pertaining to that hour of the 60 days data. Such models are generated for each of the traffic types. So an HTTP data set for 12^{th} hour has 60 days of HTTP data for the same hour. The models generated in the training phase were used for classification of the given data set during testing. Testing was performed on Internet data collected over 30 days. Data sets of 60 minutes, 30 minutes and 15 minutes were used for testing. The data collection interval used for obtaining training data is same as for obtaining test data, so that the data sets are subject to the same biases.

Classification using VQ was tested with one hour models. Bayesian classification using Gaussian mixtures was tested using models of one hour, 30 minutes and 15 minutes; and the results were compiled separately. It should be noted that the actual test data used are obtained by extracting packet trains for a connection. The testing program will predict the traffic type for the given set of packet trains. The correctness of classification is determined by comparing the predicted traffic type with the actual traffic type. Accuracy is the percentage of data sets predicted correctly using the models.

It was observed that the number of mixtures (or clusters) required to model a particular traffic class depends on the application protocol. This number is manually found out by comparing the mean and variance of a mixture with the means and variances of other mixtures. For this, the training algorithms used in both the approaches (VQ and GMM) were executed for different number of clusters/mixtures to find the optimal number of clusters/mixtures required. The number of mixtures required to model different traffic classes were found to be different. For example, HTTP traffic was modeled using 11 different Gaussian mixtures, whereas POP3 traffic required only 7 Gaussian mixtures to model it.

6.1 Results using VQ

Table 1 shows the accuracy in classifying traffic using VQ. The results of classification with and without thresholds for traffic classes are shown in Table 1(a) and Table 1(b) respectively. By defining thresholds for the distortion of each traffic type, misclassification error has reduced. The overall performance of classification increased from 83.3% accuracy to 94.9% accuracy.

Table 1. Results using VQ for one hour data

(a) Without any threshold

Traffic Type	Accuracy
HTTP	98.55%
SMTP	81.16%
DNS	100%
POP3	76.08%
SSH	60.86%

(b) With thresholds defined

Traffic Type	Accuracy
HTTP	99.27%
SMTP	96.38%
DNS	100%
POP3	90.56%
SSH	88.40%

Table 2. Results using GMM for one hour data

(a) Without any threshold

Traffic Type	Accuracy
HTTP	99.60%
SMTP	99.30%
DNS	100%
POP3	95.90%
SSH	79.18%

(b) With thresholds defined

Traffic Type	Accuracy
HTTP	99.60%
SMTP	99.30%
DNS	100%
POP3	97.20%
SSH	96.92%

6.2 Results using GMM

Table 2 shows the result of classifying traffic using the GMM. An improvement in classifying traffic (from 96.9% accuracy to 98.6% accuracy) was observed when the thresholds are defined. Also, classification using multiple parameters, namely packet train length and packet train size, has given more accurate results when compared to the previous work using Bayesian analysis techniques [8] which obtained an accuracy of 95% using 28 minutes data. Comparing tables 1 and 2 shows that modeling using GMM gives more accurate classification results than modeling using VQ. Test results in Table 2 was obtained with traffic for one hour slots. Tables 3 and 4 show results with thresholds defined, using 30 minutes data and 15 minutes data respectively.

6.3 Discussion

It is evident from the test results that the DNS traffic is always identified correctly. DNS is an Internet service which uses UDP as the underlying transport protocol. Since UDP is a connectionless protocol, each DNS query or reply will be a packet train with single packet. This clearly distinguishes it from TCP flows which will normally have more than one packet in a train. Due to this unique packet train property of the DNS traffic, it is never misclassified.

Tables 2, 3 and 4 show that HTTP and SMTP traffic types were classified with almost the same accuracies for different time durations even without thresholds. The mixtures of HTTP traffic type did not overlap with mixtures of any other traffic types resulting in highly accurate classification. SMTP and POP3 traffic had two overlapping mixtures; therefore POP3 got misclassified as SMTP

Table 3. Results using GMM for 30 minutes data with thresholds defined

Traffic Type	Accuracy
HTTP	99.84%
SMTP	99.68%
DNS	100%
POP3	96.99%
SSH	96.72%

Table 4. Results using GMM for 15 minutes data with thresholds defined

Traffic Type	Accuracy
HTTP	99.78%
SMTP	99.67%
DNS	100%
POP3	96.28%
SSH	94.53%

traffic. This is expected as both POP3 and SMTP deal with mail traffic. The characteristics of both types of traffic are expected to be similar.

One important feature of these models generated using packet train parameters is that it can be used by network administrators to estimate the number of *tiny* flows, that is ones that involve only a small number of packets. Similarly, administrators can also monitor *heavy hitters* [21], those that represent a significant proportion of the traffic, or the link bandwidth. For example, consider some of the relevant HTTP mixtures for inbound packet trains during 11-12 hour from the TeNeT link, which are shown in table 5. The table shows that most of the HTTP downloads are small packet trains. As seen in rows 1, 3, and 4 of the table, more than 50% of downloads are trains consisting of 5 packets, with the download size ranging approximately from 400 to 1000 bytes. Near to 12% of download traffic are medium-size downloads as seen in row 5, and more than 1% are huge downloads as depicted in the last row of the table. Such information can be used to learn the different classes of service required for various applications in the network. The knowledge of traffic characteristics enables an administrator to do QoS provisioning and also to prioritize different traffic classes depending on the QoS requirements.

This statistical method also reveals typical use of an application. For example, if HTTP were used for relaying some other traffic (such as SSH) or for streaming large amount media traffic, rather than more traditional web browsing. Also traffic traces were collected during file transfer between machines using SCP, which uses the same port as SSH (port 22). The SCP packet trains generated by transferring files of size greater than 2 Megabytes was successfully rejected by the SSH model. Though SCP and SSH use the same port, SSH traffic consists of small packets (of size between 100 bytes to 200 bytes), whereas SCP traffic usually has large packets (of size usually greater than 1000 bytes).

Table 5. Relevant HTTP mixtures during 11:00-12:00

Index	Packet train length		Packet train size		Percentage of total trains
#	μ	σ	μ	σ	%
1.	5.10	1.02	854.52	188.30	20.24
2.	10.54	3.0	7428.57	4485.69	18.85
3.	4.81	0.41	576.14	27.58	18.27
4.	4.67	0.67	436.87	39.34	13.00
5.	34.56	18.02	37356.1	24876.7	11.90
6.	6.06	0.74	2644.25	545.22	6.93
7.	5.21	1.11	1522.81	163.07	5.51
8.	3.56	0.97	243.79	92.40	2.70
9.	520.56	1582.78	619126.8	2105716.6	1.18

7 Conclusion

Our experiments show that GMM can be used to generate better models as compared to VQ. The accuracy of Bayesian classification for one hour data using GMM is 98.6% compared to 94.9% achieved using VQ based classification. Accuracy of classification using GMM for data of 15 minutes is 98.05%. The significance of time duration is that, only 15 minutes of data is required to identify a particular traffic type. Although VQ and GMM have been used for modeling of network traffic in the past [8, 10], a novelty of the work presented in this paper is the use of new parameters, namely *packet train length* and *packet train size*. Using *packet train length* and *packet train size* as multiple parameters for modeling, has helped not only in yielding high accuracy in classification, but also in revealing useful information on different service classes in network traffic.

This work looked only at one UDP based traffic class, namely DNS traffic. But, it should be noted that almost all UDP based applications can be viewed as a connection oriented traffic and therefore can be represented using packet trains. For example, a video conferencing tool running on top of UDP can be identified uniquely by *src host*, *src port*, *dst host* and *dst port*, and all such packets can be considered as part of a connection. To distinguish different connections of the same applications, the time between packets can be used; as the time between consecutive packets of one run of the application will be much less as compared to time between consecutive packets of different runs of the application. Hence, using all these information, packet train parameters can be extracted for the traffic generated by a video conferencing application, or in general, for traffic generated by almost any UDP based application.

References

1. Croll, A., Packman, E.: Managing Bandwidth: Deploying Across Enterprise Networks. Prentice Hall PTR Internet Infrastructure Series. Prentice Hall (2000)

2. Sally Floyd and Vern Paxson: Difficulties in simulating the Internet. IEEE/ACM Trans. on Networking **9**(4) (2001) 392–403
3. Logg, C.: Characterization of the traffic between SLAC and the Internet. *http://www.slac.stanford.edu/comp/net/slac-netflow/html/SLAC-netflow.html* (2003)
4. Moore, A.W., Papagiannaki, K.: Toward the accurate identification of network applications. In: Passive and Active Network Measurement, Sixth International Workshop, PAM 2005. (2005) 41–54
5. Roughan, M., Sen, S., Spatscheck, O., Duffield, N.G.: Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification. In: Internet Measurement Conference, IMC '04. (2004) 135–148
6. McGregor, A., Hall, M., Lorier, P., Brunskill, J.: Flow Clustering Using Machine Learning Techniques. In: Passive and Active Network Measurement, Fifth International Workshop, PAM 2004. (2004) 205–214
7. Chen, Y.W.: Traffic behavior analysis and modeling of sub-networks. International Journal of Network Management **12**(5) (2002) 323–330
8. Moore, A.W., Zuev, D.: Internet traffic classification using bayesian analysis techniques. In: Proc. of the 2005 ACM SIGMETRICS, International Conference on Measurement and Modeling of Computer Systems, ACM Press (2005) 50–60
9. Saifulla, M.A., Murthy, H.A., Gonsalves, T.A.: Identifying Patterns in Internet Traffic. In: International Conference on Computer Communication. (2002) 859–865
10. M. A. Saifulla: A Pattern Matching Approach To Classification Of Internet Traffic. Master's thesis, Indian Institute of Technology, Madras (2003)
11. Tamir, D., yeon Park, C., Yoo, W.S.: Vector Quantization and Clustering: A Pyramid Approach. IEEE Data Compression Conference and Industrial Workshop (1995)
12. Roberts, S.J., Husmeier, D., Penny, W., Rezek, I.: Bayesian Approaches to Gaussian Mixture Modelling. IEEE Trans. on Pattern Analysis and Machine Intelligence **20**(11) (1998) 1133–1142
13. Stevens, W.R.: TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley (1994)
14. IANA: Internet Assigned Numbers Authority. *http://www.iana.org/assignments/port-numbers* (2006)
15. Claffy, K.C.: Internet traffic classification. PhD thesis, University of California, San Diego (1994)
16. Paxson, V.: Empirically derived analytic models of wide-area TCP connections. IEEE/ACM Trans. on Networking **2**(4) (1994) 316–336
17. Jain, R., Routhier, S.: Packet Trains-Measurements and a New Model for Computer Network Traffic. IEEE Journal on Selected Areas in Communications **SAC-4**(6) (1986) 986–995
18. M.Gray, R.: Vector Quantization. IEEE ASSP Magazine **1** (1984) 4–29
19. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Second edn. Wiley-Interscience Publication (2001) Chapter 2.
20. TeNeT: The Telecommunications and Computer Networking Group, Indian Institute of Technology, Madras. *http://www.TeNeT.res.in* (1996)
21. Graham Cormode and Flip Korn and S. Muthukrishnan and Divesh Srivastava: Diamond in the Rough: Finding Hierarchical Heavy Hitters in Multi-Dimensional Data. In: Proc. of the ACM SIGMOD, International Conference on Management of Data. (2004) 155–166